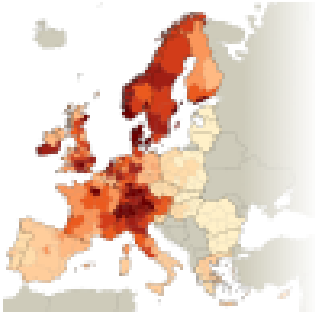


HyperAdmin User Manual



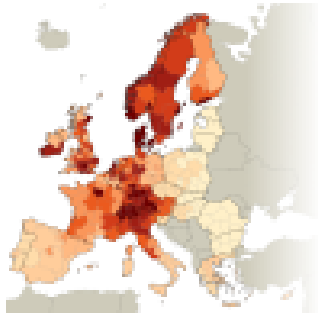
HyperCarte Research Gr



<http://hypercarte.imag.fr>

HyperCarte Research Group

HyperAdmin



HyperCarte Research Group

User

Manual:

HyperCarte Research Gr



<http://hypercarte.imag.fr>

Abstract

This document provides the reader with some information on how to prepare the set of input documents for the generation of an HyperAtlas dataset `hyp` file with the HyperAdminimal software.

Table of Contents

Preface	vi
1. HyperAdmin Overview	1
2. How to use HyperAdminimal	3
2.1. Startup	3
2.2. Fill Input Form	4
2.3. Save and Load Build Parameters	6
2.4. Run	7
2.4.1. Known Problems	9
3. Geometry input	11
3.1. From Shapefile to MIF MID	11
3.2. The MID file	12
3.3. The MIF file	13
3.4. Layer of main cities	16
4. Structure/Contiguity input	18
4.1. Structure input	18
4.2. Contiguity input (optional)	22
5. Stocks input	28
5.1. HyperAdmin input data file format	28
5.1.1. About	28
5.1.2. Data	28
5.1.3. Default	29
5.1.4. Label	29
5.1.5. Metadata	30
5.1.6. Provider	30
5.1.7. RatioStock	30
5.1.8. StockInfo	31
A. References	33
About	xxxiv

List of Figures

1.1. HyperCarte Workflow	1
2.1. HyperAdminimal Startup Console	3
2.2. HyperAdminimal Startup GUI	4
2.3. Build Parameters XML File	6
2.4. Previous Build Parameters XML File	7
2.5. Confirm Overwrite Window	7
2.6. Dataset Build Result Dialog	8
2.7. Dataset Build Console Output Example	9
3.1. QGIS: Save as Mapinfo MIF Format	12
3.2. MIF file header	13
3.3. Example of two "Region" entries in the MIF file Data section	15
4.1. Number S of needed sheets for n units	27

List of Tables

4.1. Overview of expected sheets for data structure input	19
4.2. Sample input Unit sheet	19
4.3. Sample input Area sheet	20
4.4. Sample input Zoning sheet	20
4.5. Sample input UnitSup sheet	20
4.6. Sample input UnitArea sheet	20
4.7. Sample input UnitZoning sheet	21
4.8. Sample input Language sheet	21
4.9. Sample input UnitLanguage sheet	21
4.10. Sample input AreaLanguage sheet	21
4.11. Sample input ZoningLanguage sheet	22
4.12. Overview of expected sheets for contiguity input	22
4.13. Sample input Contiguity sheet	23
4.14. Sample input ContiguityLanguage sheet	23
4.15. Sample input Neighbourhood sheet	24
4.16. Sample input NeighbourhoodLanguage sheet	25
4.17. Sample input ContiguityZoning sheet	26
4.18. Sample input ContiguityArea sheet	26
4.19. Sample input UnitContiguity sheet	26
4.20. Example of needed sheets number	27
5.1. V2 sample About sheet	28
5.2. V2 sample Data sheet	29
5.3. V2 sample Default sheet	29
5.4. V2 sample Label sheet	29
5.5. V2 sample Metadata sheet	30
5.6. V2 sample Provider sheet	30
5.7. V2 sample RatioStock sheet	31
5.8. V2 sample StockInfo sheet	31

Preface

HyperCarte Research Group

HyperCarte Research Group aims at providing projects and applications for interactive cartography. The projects focus on the development of an easily understood methodology that allows the analysis and visualization of spatial phenomena, taking into account its multiple possible representations.

Statistical observations of the territory are complex, and one representation, directly linked to a precise objective, is the result of a combination of different choices which are relative on one hand to the territories and their geographical scales, to the the statistical indicators on the other hand. This is of interest for researchers as well as for development policy decision-makers.

Thus, the principal innovative aspect of the HyperCarte project lies on this perspective based on the popularization of methods coming from spatial analysis such as the fitting of territorial scales, gradients, discontinuities.... This supposes an effort of multidisciplinary cooperation between geographers and computer scientists in order to create new maps in real time according to the different choices. An important effort has concerned ergonomics and time of calculus.

Partners of the HyperCarte Research Group are the following teams:

 RIATE

UMS 2414 RIATE <http://www.ums-riate.com>



CNRS UMR 8504 G  ographie-Cit  s [UMR 8504]
<http://www.parisgeo.cnrs.fr>

 Mescal

LIG Mescal [UMR 5217] <http://mescal.imag.fr/>



LIG STEAMER [UMR 5217] <http://steamer.imag.fr/>

For further information, please visit HyperCarte Research Group Web site on <http://hypercarte.imag.fr>. For any question comment or suggestion, please contact the HyperCarte Research Group at [<hypercarte@imag.fr>](mailto:hypercarte@imag.fr).

HyperAdmin User Manual

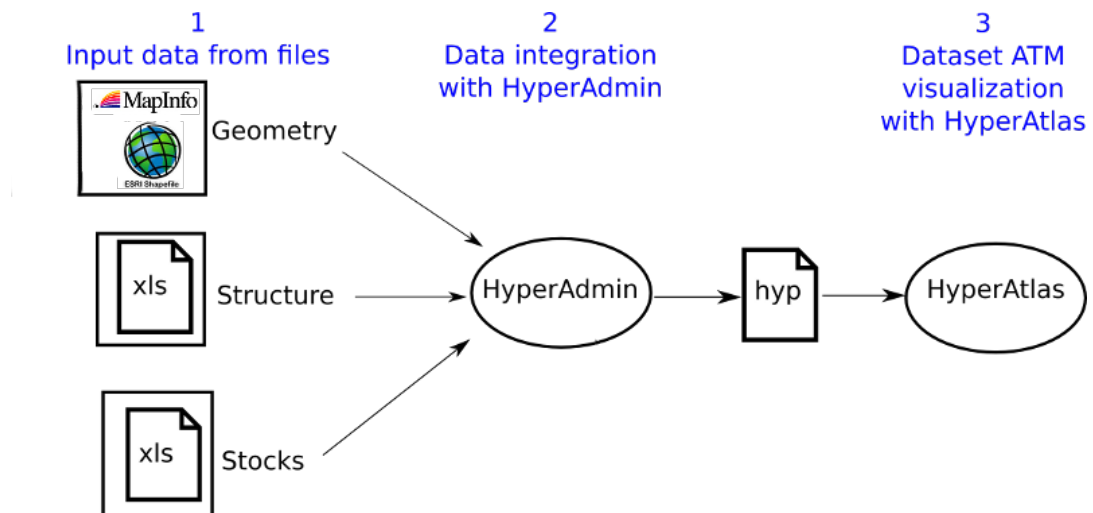
This document proposes guidelines on how to build HyperAtlas dataset .hyp files with the latest version of the HyperAdmin software named **HyperAdminimal**. This software has been designed and improved on the occasion of the HyperAtlas/HyperAdmin customized distribution for the project named *Politiques de la Terre    l'usage de l'Anthropoc  ne* (July 2015). For further information, please consult hyperpolter [<http://hyperpolter.forge.imag.fr>] project Web Site.

Chapter 1. HyperAdmin Overview

In order to perform Multiscalar Territorial Analysis with HyperAtlas, the datasets are serialized in a convenient format into a binary file named with the `.hyp` extension. As a convention, an HyperAtlas dataset is called an **hyp file** (example: `demography.hyp`).

HyperAdmin is the tool to generate hyp files from a set of input well-formed files. The steps to generate an hyp file and the workflow between HyperAdmin and HyperAtlas is summarized in the Figure 1.1.

Figure 1.1. HyperCarte Workflow



HyperAdmin and HyperAtlas data flow.

To sum up, the main expected input files are:

- the **geometry** of the dataset, in MapInfo MIF/MID format (historical supported format) or ESRI Shapefile format (since July 2015).
- the **structure** of the dataset, as an `xls` (Excel/OpenOffice) file.
- the **stocks** of the dataset, as an `xls` (Excel/OpenOffice) file

As shown on Figure 1.1, creating a dataset hyp file consists in:

1. preparing your dataset geometry;
2. preparing your dataset structure as a spreadsheet `structure.xls` file;
3. optionally, preparing a distance-time matrix as an `xls` file for custom contiguities;
4. preparing your dataset stocks as a spreadsheet (Excel/OpenOffice) `data.xls` file;
5. generating the dataset hyp file with HyperAdmin.

The following chapter describes each above step for integrating your data into a new dataset hyp file.

Most of users should use a pre-defined set of input files: geometry structure and data templates and ready to use examples are available and regularly delivered by the HyperCarte Research Group. Latest dataset hyp files have been generated with them.

The following chapter targets such users so they can quickly generate a new dataset hyp file with HyperAdminimal software.

HyperAdmin Overview

From Stocks input chapter, more details are given about the content and constraints of the input files.

Chapter 2. How to use HyperAdminimal

2.1. Startup



Requirements

To execute HyperAdminimal, a Java Runtime Environment (version 1.7 or higher) is required on your environment.

To check this requirement, open a terminal console (dos/shell) and type the **java -version** command. If your environment is Java-enabled, a message indicating the version (as shown below: 1.7.0_51) must be displayed:

```
$ java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)
```

For more information on how to install a Java Runtime Environment, please consult <http://java.com>.

This HyperAdminimal 1.0.2 distribution also requires a graphical environment.

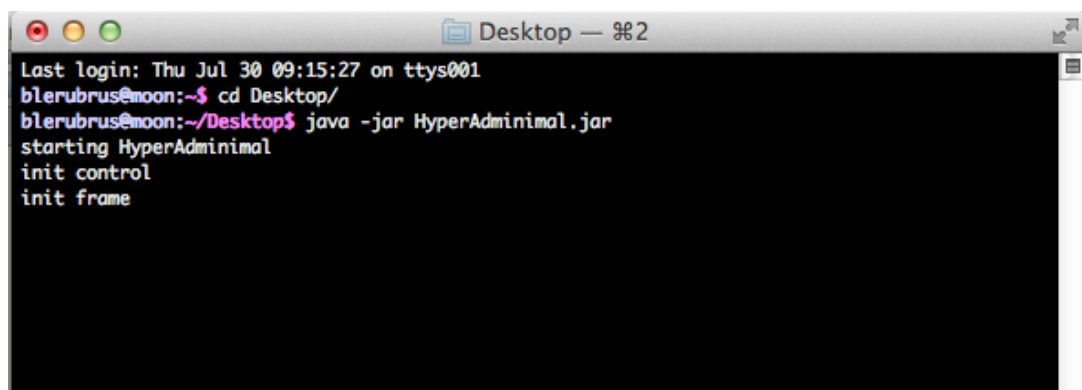
The HyperAdminimal application is delivered as a Java executable binary file named `hyperadminimal.jar`.

To execute HyperAdminimal software, open a console (depending on your platform: Windows-dos / Mac-Terminal / Linux-shell) and type the following command from the directory where `hyperadminimal.jar` is located on your disk:

```
java -jar hyperadminimal.jar
```

Figure 2.1 and Figure 2.2 respectively show the output messages displayed to the console and the expected Graphical User Interface (GUI) main frame of the application.

Figure 2.1. HyperAdminimal Startup Console



This screenshot shows the console output messages when executing HyperAdminimal. In this use case, the `HyperAdminimal.jar` file is located in the user's `~/Desktop/` folder.

Figure 2.2. HyperAdminimal Startup GUI

The screenshot shows the HyperAdminimal application window. At the top is a title bar with the name 'HyperAdminimal' and standard window control buttons (red, yellow, green). Below the title bar is a toolbar with four icons: a question mark (help), a folder with a plus sign (file selection), a floppy disk (save), and a play button. The main content area is a form with the following fields and controls:

- Geometry Format:** Two radio buttons. 'ESRI Shapefile' is selected (indicated by a filled circle), and 'MapInfo MIF/MID' is unselected (indicated by an empty circle).
- Geometry File:** A text input field followed by a blue button with a folder icon.
- Property Name (shp only):** A single-line text input field.
- Structure xls File:** A text input field followed by a blue button with a folder icon.
- Stocks (data) xls file:** A text input field followed by a blue button with a folder icon.
- Contiguity (optional):** A text input field followed by a blue button with a folder icon.
- Cities Layer (optional):** A text input field followed by a blue button with a folder icon.
- Author Name:** A single-line text input field.
- Author Firstname:** A single-line text input field.
- Dataset Name:** A single-line text input field.
- Dataset Description:** A single-line text input field.
- Output File:** A text input field followed by a blue button with a folder icon.

HyperAdminimal window at startup.

2.2. Fill Input Form

As shown in Figure 2.2, the application is composed of a basic toolbar and of a form composed of input text fields.

The text fields aim at specifying the input parameters for the generation of the dataset .hyp file. The

📁 buttons on the right side of the window helps at selecting the input (or output) files on your disk with a file chooser, though any absolute path to file may be manually written in the text field.

Note that most of field values are required except `Property Name` (mandatory only for the ESRI Shapefile geometry input format option), `Contiguity` and `Cities Layer`.

Please find below a short description of the form fields:

- **Geometry Format**

This radio button aims at specifying the format of your input geometry among the ESRI Shapefile or the MapInfo MIF/MID options.

- **Geometry File**

Depending on the **Geometry Format** previously selected option:

- **Shapefile** option: expects the absolute path on your disk to a valid .shp file. Note that a valid Shapefile is composed of several files with the same basename, but at least three files with the same basename must exist and be located in the same directory with the following extensions: .shp, .dbf and .shx. Choose the .shp for the value of the **Geometry File** field.
- **MIF/MID** option: expects the absolute path on your disk to a .mif or .mid file. Note that both .mif and .mid files must exist in the same directory with the same base name, for example, "nuts2010_eu31.mif" and "nuts2010_eu31.mid". Choose one or the other filename for the **Geometry File** field value. See also Geometry input for the expected content.

- **Property Name**

This field is mandatory for the ESRI Shapefile input geometry format. The value in this field indicates the name of the property (shapefile attribute) to use for spatial unit codes. Indeed, many properties may be included in a Shapefile (typical property names are id, the_geom, for example). HyperAdmin parses the shapefile and creates a map of spatial unit codes / geometry pairs. Each geometry has to be attached to a spatial unit code. The unit codes must match the ones used in the structure and data xls files. Please fill this field with the name of the shapefile property whose values match the spatial unit codes in the structure (and data) xls files. See also Section 2.4.1.1 to get the possible values of the available property names in the Shapefile.

- **Structure**

The value of this field is the absolute path on your disk to the dataset input structure xls file. Please consult Structure/Contiguity input for further information.

- **Stocks**

The value of this field is the absolute path on your disk to the dataset input data (aka stocks) xls file. Please consult Stocks input for further information.

- **Contiguity**

The value of this optional field is the absolute path on your disk to the dataset custom contiguity definition xls file. Please consult Section 4.2 for further information.

- **Cities Layer**

The value of this optional field is the absolute path on your disk to a csv file providing the coordinates of main cities that will be displayed as an additional layer in HyperAtlas. Please consult Section 3.4 for further information.

- **Author Name**

Please type your name in this field (or the name of your organization).

- **Author Firstname**

Please type your firstname in this field.

- **Dataset Name**

Please type a pretty name for your dataset.

- **Dataset Description**

Please type a description for your dataset, for example, an abstract or the topic of this hyp file to be generated (demography, economy, land-use, ...).

- **Output File**

Type the absolute path on your disk to the dataset hyp file to be generated. Example: /home/user/myNewDataset.hyp



Please insure to add the .hyp extension to this output filename. For compatibility reasons, please avoid any accentuated or blank characters for this filename.

2.3. Save and Load Build Parameters

In order to replay several times a dataset build, the HyperAdminimal application proposes a functionality to save and load the build parameters. This fonctionnality can be explained by describing the available buttons on the toolbar:



Reset parameters

This button erases the values in the fields.



Load parameters

Clicking this button opens a file chooser, the user is invited to select on his/her disk an XML file that has previously been saved via the save parameters [6] fonctionnality.



Save parameters

On clicking this button, the user is invited to select a directory and a filename on his/her disk. The content of the fields will be saved to this file in XML format. As shown in Figure 2.3, the July 2015 format of this XML file is quite simple, it can thus be eventually easily copied/pasted and manually edited for various configurations (see also the older xml format in Figure 2.4). Nevertheless, please note that this xml file can not easily be shared among several users on different workstations, since the paths to files are given with an absolute path.

Figure 2.3. Build Parameters XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<hyperadmin>
  <geoformat>SHAPEFILE</geoformat>
  <geofile>/Users/blerubrus/workspace/hypercarte/dataset/pdlt/Communes_IDF_Igr
  <shapefileUnitCodePropertyName>CODE_COM</shapefileUnitCodePropertyName>
  <structure>/Users/blerubrus/workspace/hypercarte/dataset/pdlt/structure.xls<
  <data>/Users/blerubrus/workspace/hypercarte/dataset/pdlt/stocks.xls</data>
  <contiguity />
  <authorname>Le Rubrus</authorname>
  <authorfirstname>Benoit</authorfirstname>
  <datasetname>testShapefile</datasetname>
  <datasetdescription>test with shapefile input format</datasetdescription>
  <output>/Users/blerubrus/testShapefile.hyp</output>
</hyperadmin>
```

Figure 2.4 shows the previous format of the xml file (before July 2015). Please note that this old format is still supported by HyperAdminimal, though the new format (since July 2015) can not be used with older versions of HyperAdminimal.

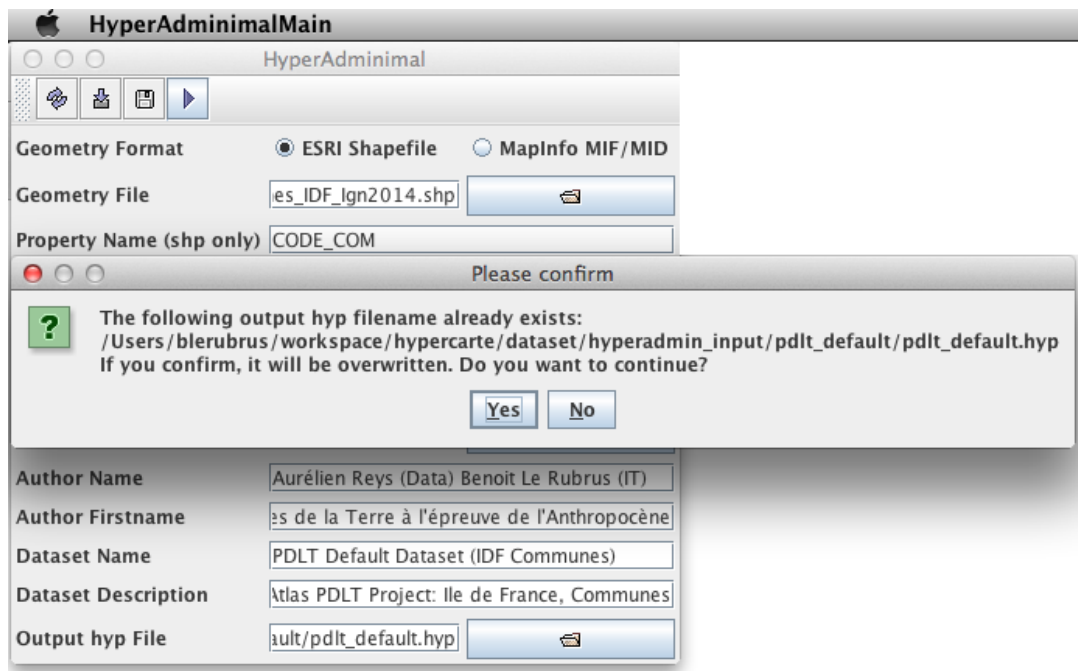
Figure 2.4. Previous Build Parameters XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<hyperadmin>
  <mif>/home/blerubrus/geom_nuts2_eu31.mif</mif>
  <mid>/home/blerubrus/geom_nuts2_eu31.mid</mid>
  <structure>/home/blerubrus/structure.xls</structure>
  <data>/home/blerubrus/data.xls</data>
  <contiguity />
  <authorname>HyperCarte Research Group</authorname>
  <authorfirstname>LIG STEAMER</authorfirstname>
  <datasetname>testDataset</datasetname>
  <datasetdescription>Built with HyperAdminimal</datasetdescription>
  <output>/home/blerubrus/test.hyp</output>
</hyperadmin>
```

2.4. Run

Once all required fields are filled, simply start the build of the dataset hyp file on clicking the "Play" button of the toolbar.

As shown in Figure 2.5, a dialog box may appear to inform you that you are about to overwrite an already existing dataset hyp file.

Figure 2.5. Confirm Overwrite Window

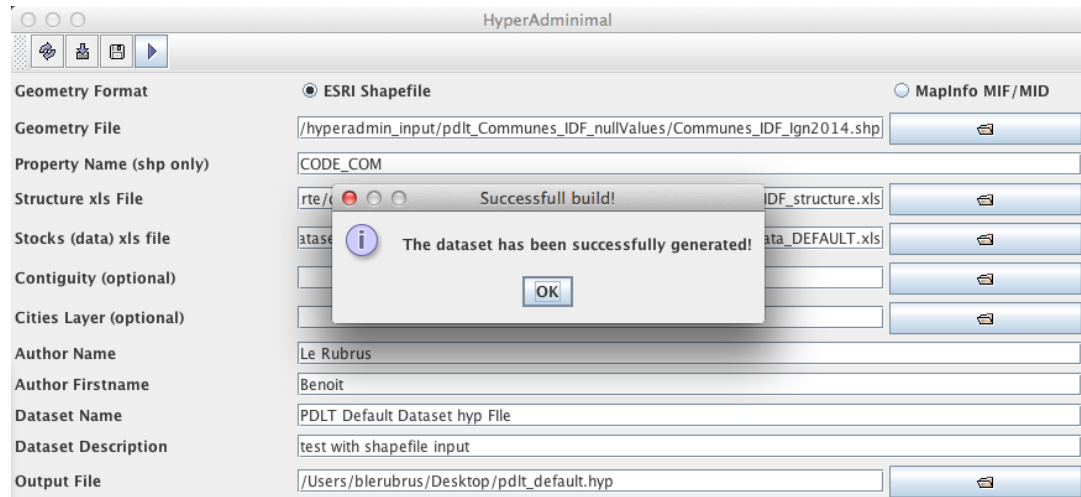
This dialog appears when you are about to start the build of a dataset hyp file to an already existing file. Click "No" to abandon and change your output hyp filename.

For information, most of dataset hyp files are build in between 1 and 10 minutes, depending on the number of spatial units and of indicators.

Build log messages are displayed to the console in which HyperAdminimal has been executed (see the command to execute HyperAdminimal[3]). When things go wrong, these build log messages help at fixing the input files.

At the end of the process, a dialog box over the main window displays a message that indicates if the build failed or succeeded, as shown in Figure 2.6.

Figure 2.6. Dataset Build Result Dialog



In case of a successful dataset hyp file generation, at the end of the process, hyperadminimal displays such a dialog box over the main window.

Besides, the console output (an example is given in Figure 2.7) proposes the full build log information and, in the better cases, ends with the following message : *The dataset has been successfully generated into the (...).hyp file.*

Figure 2.7. Dataset Build Console Output Example

```
Pending progress: 40 % (stock-time step completed)
Pending progress: 47 % (unit-stock step completed)
Pending progress: 60 % (aggregation step completed)
-----Pending progress: 98 % (no additional layers)
100 % completed - Process has completed

Map bounds ok
1300 Units in MIF MID
1308 Units descriptions in data source
1 descs for 1 areas... Areas ok
8 Unit-Area relationships hierarchy
2 Zonings
1308 Unit-Zoning relationships
5 Stocks
4 Relevant ratios
6500 Unit-Stocks values
1300 Hierarchy ok
Aggregation ok
Neighbourhood ok
Processing simple contiguity...
Contiguity computing begins
Using JTS to avoid time consuming database queries
Computed 100% (more exactly 8/8) of the units contiguities for zoning rank 1 in
Using JTS to avoid time consuming database queries
Computed 7% (more exactly 100/1,300) of the units contiguities for zoning rank
Computed 15% (more exactly 200/1,300) of the units contiguities for zoning rank
Computed 23% (more exactly 300/1,300) of the units contiguities for zoning rank
Computed 30% (more exactly 400/1,300) of the units contiguities for zoning rank
Computed 38% (more exactly 500/1,300) of the units contiguities for zoning rank
Computed 46% (more exactly 600/1,300) of the units contiguities for zoning rank
Computed 53% (more exactly 700/1,300) of the units contiguities for zoning rank
Computed 61% (more exactly 800/1,300) of the units contiguities for zoning rank
Computed 69% (more exactly 900/1,300) of the units contiguities for zoning rank
Computed 76% (more exactly 1,000/1,300) of the units contiguities for zoning rank
Computed 84% (more exactly 1,100/1,300) of the units contiguities for zoning rank
Computed 92% (more exactly 1,200/1,300) of the units contiguities for zoning rank
Computed 100% (more exactly 1,300/1,300) of the units contiguities for zoning rank
Computed 100% (more exactly 1,300/1,300) of the units contiguities for zoning rank
Contiguity computing has achieved in 9,681 milliseconds
Simple contiguity ok
Saving simple contiguity...
no additional map layers
The dataset has been successfully generated into the pdlt_default.hyp file
```

If the process does not finish after a reasonable duration (the largest dataset hyp file ever created with HyperAdminimal took 15 minutes long), close the terminal console and restart hyperadminimal.

2.4.1. Known Problems

2.4.1.1. Empty Property Name field

As described in Section 2.2, the **"Property Name"** field is mandatory to integrate a Shapefile. Here is an example of log output when this field is empty when attempting to integrate a dataset hyp file:

How to use HyperAdminimal

The given simple feature property name '' could not be found in the shapefile.
Available property names are:

```
'the_geom' ;  
'CODE_COM' ;  
'LIB_COM' ;  
'CODE_ZE' ;
```

(...)

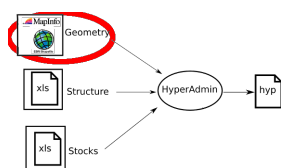
java.lang.IllegalArgumentException: The given simple feature property name ''
could not be found in the shapefile.

(...)

In human-readable language, HyperAdminimal could not build with the given "empty" property name... Fortunately, it also returns the list of the possible property names for this shapefile: the_geom, CODE_COM, etc. Consequently, forcing the error and exception trigger can be useful to know which property name to indicate in the HyperAdminimal form field.

In such a case, please note that HyperAdminimal must be restarted.

Chapter 3. Geometry input



This section describes the expected geometry input by HyperAdmin.

The HyperAtlas maps are computed by aggregating the geometric information at the lowest level of the mesh hierarchy. For example, for a structured hierarchy made of NUTS at levels 0 (country), 1 (big regions), 2 (similar to french administrative regions) and 3 (french administrative departments), the user must provide HyperAdmin with the geometry of the finest territorial units, that is to say at NUTS level 3.

The main difficulty when building a dataset hyp file is to provide HyperAdmin with a correct geometry, without any hole, and a perfect topology between the contiguous territorial units.

Since July 2015, HyperAdmin has supported the ESRI Shapefile format as geometry input. Historically, the expected geometry used to be provided by the user in the Pitney Bowes MapInfo MIF/MID format. For further information about these geometry formats, please consult:

- About Shapefile: see [7] and [2].
- About MIF/MID: see [4] and <http://www.mapinfo.com/> (last visit: 27th of July 2015).

The following sub-sections describe the HyperAdmin expected geometry input in the MapInfo MIF/MID format.

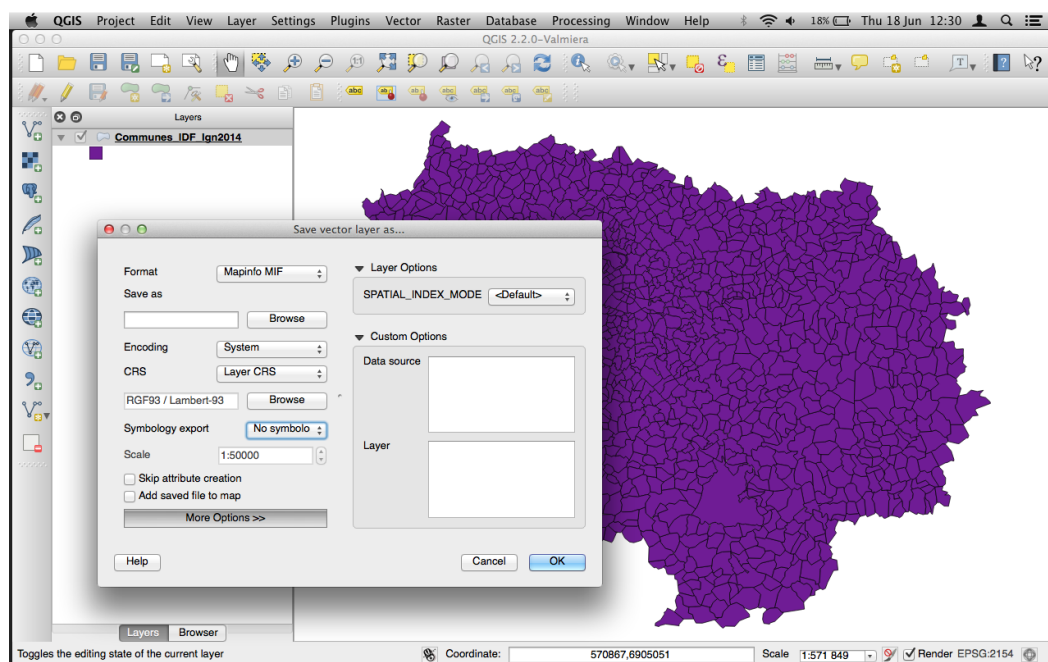
3.1. From Shapefile to MIF MID

Historically, the generation of the dataset hyp files has been designed to use the MapInfo MifMid format as input geometry. On the occasion of the *HyperAtlas Politiques de la Terre* project (2015), the HyperCarte Research Group decided to support the ESRI Shapefile format as input geometry. Though popular since many years, most of the existing dataset hyp files have been created using the MIF MID format, most of the time by converting a given geometry in Shapefile format to this MIF MID format, for the needs of HyperAdmin.

To be short, this section proposes a simple solution to convert a geometry in Shapefile format to a geometry in MIF MID format:

- If not done yet, download Quantum GIS [6].
- From QGIS, open your Shapefile as a Vector Layer.
- Save your vector Layer as "Mapinfo MIF" format. QGIS proposes a dialog box (shown in Figure 3.1) whose default parameters should be enough to get a geometry in MIF MID format.

Figure 3.1. QGIS: Save as Mapinfo MIF Format



This QGIS screenshot shows the dialog box to save a vector layer into the MapInfo MIF/MID format. The background window shows the previously opened Shapefile Vector layer. Simply click the "Browse" button to select a target directory for your MIF/MID then click "OK".

3.2. The MID file

The MID file must be made of only one column where territorial units identifiers are listed, one per line, without any doublon. Example:

```
"AT111 "
"AT112 "
"AT125 "
"AT126 "
"AT127 "
"AT13 "
"AT211 "
```

! The given order of TU identifiers in the MID file must match the order of provided regions in the MIF file, see Data section of the MIF file [15]

Based on a naming convention of the identifiers for these territorial units, following exceptions are handled by HyperAtlas for particular display options. Please take into account the following exceptions when designing your dataset:

- **FR, ES, PT, MT** is the list of units identifiers for countries that own overseas units: France (Martinique, ...), Spain (Canarias, ...) and Portugal (Madeira). For example for European datasets, In HyperAtlas, the islands will be drawn in squares over the Russia.
- **SUR** and **BRA** (Surinam and Brazil) are examples of units identifiers that are treated differently when drawing them on the maps by HyperAtlas.
- Integer identifiers from **0** to **9** correspond to squares that must be drawn on the map, they are used for overseas in Europe dataset.

- A territorial unit with the identifier **no data** will be painted in white on the maps that are drawn by HyperAtlas. This exception is used for North Cyprus in Europe datasets.
- The **chypre** identifier is used to handle the particular case of the display of Cyprus island in the ESPON datasets.

3.3. The MIF file



The information in this section is essentially based on the MapInfo Data Interchange Format document [4].

Geographical units are described in an ASCII file by their X and Y coordinates. The .MIF file is made of an header section then a data section.

Figure 3.2. MIF file header

```
VERSION n
Charset "characterSetName"
[ DELIMITER "<c>" ]
[ UNIQUE n,n.. ]
[ INDEX n,n.. ]
[ COORDSYS... ]
[ TRANSFORM... ]
COLUMNS n
  <name> <type>
  <name> <type>

( ... )
```

As shown on Figure 3.2, the header can contain the following information:

- **VERSION**: the version of the MapInfo software;
- **CHARSET** clause specifies which character set was used to create text in the table (examples: WindowsLatin1, MacRoman or Neutral);
- **DELIMITER** shows the character that is used to separate columns values (if not specified, tabulation is the default delimiter);
- **UNIQUE** parameter must be a number that refers to a database column, this parameter is used to create related tables;
- **INDEX** parameter (a number or a comma-separated list of numbers) that shows the number(s) of the indexed column(s);
- the **COORDSYS** parameter sets the used coordinate system.

This parameter is essential, in particular to compute the scale of the map. By default (when no COORDSYS clause is specified) data is assumed to be stored in longitude/latitude forms. All coordinates are stored with respect to the northeast quadrant. The coordinates for points in the West of Greenwich have a negative X while coordinates for points in the East of Greenwich have a positive X. Coordinates for points in the Northern hemisphere have a positive Y while coordinates for points in the Southern hemisphere have a negative Y. Examples:

- The following example represents a map of Europe centered on 50°N 15°E with a Lamber Azimutal projection that can be associated to the following bounds pair: (X_{min}, Y_{min}) (X_{max}, Y_{max}). The "m" option stands for "meters" as the unit:

```
CoordSys NonEarth Units "m" Bounds (-2217175, -1723801) (1783333, 2518193)
```


- Another setting for a map of Rhône-Alpes may be:

```
CoordSys NonEarth Units "m" Bounds (691594, 1893320) (993392, 2185448)
```

- **TRANSFORM** parameter can be used to convert coordinates which are given in a different quadrant than the default northeast one.
- **COLUMNS** parameter describes the data in the table of the associated MID file. The *n* parameter specifies the number of columns. Example:

```
Columns 1
    unit Char(100)
```

specifies one column named *unit*, each value will be made of characters string type whose length is not longer than 100.



HyperAdmin is quite sensible on the format of the header of the MIF file (one information by line). Here are some examples of the expected formats for the header of the more frequently recent and used MIF files:

- European datasets (used for ESPON HyperAdmin maps with the EPSG 3035 Coordinates system and projection):

```
Version 300
Charset "Neutral"
Delimiter ","
CoordSys NonEarth Units "m" Bounds (2600301.93555, 1249109.375) (6593124.
Columns 1
    ID Char(50)
Data

(...)
```

- EUROMED dataset:

```
Version 300
Charset "WindowsLatin1"
Delimiter ","
CoordSys NonEarth Units "m" Bounds (-4487557.26071, -3722255.38453) (4487
Columns 1
    ID Char(10)
Data

(...)
```

- Metroborder dataset:

```
VERSION 300
Charset "WindowsLatin1"
DELIMITER ";"
COORDSYS NonEarth Units "m" Bounds (-743051.308162917,-145654.445989655)
COLUMNS 1
    SHN Char(14)
DATA

(...)
```


The **DATA** keyword specifies both the end of the header of the MIF file and the start of the enumeration of outlines.

If the MapInfo MIF file may set different types of graphical primitives (point, line, polyline, etc.), the HyperAdmin software only expects the polygon type in order to describe the outlines of territorial units. Each TU whose identifier is given in the MID file (see Section 3.2) must be associated to a new entry in the MIF file under the **data** section, IN THE SAME ORDER, as a **Region** entry. In MapInfo, a Region object consists of one or more polygons. Let us describe an expected **Region** entry using the definition example shown on Figure 3.3.

Figure 3.3. Example of two "Region" entries in the MIF file Data section

```
Data
Region 2 ❶
  7 ❷
108071.871 -293320.749
96339.456 -282096.297
102833.097 -261179.193
106485.534 -258631.56
123883.98 -262981.491
122621.886 -282959.13
108071.871 -293320.749
  Pen (1,2,0) ❸
  Brush (0,1)
  Center 110111.718 -275976.153
  5 ❹
-407753.01 -311500.065
-417000.993 -311417.496
-411718.965 -289228.641
-406514.985 -302217.573
-407753.01 -311500.065
  Pen (1,2,0)
  Brush (0,1)
  Center -411757.989 -300364.353
Region 1 ❺
  11 ❻
2186917.593 -1518464.703
2186829.009 -1692861.786
2129979.423 -1729141.275
1933829.46 -1729141.275
1928265.747 -1699690.677
1922979.324 -1671615.192
1928499.903 -1666190.274
1941660.768 -1656068.01
2005909.794 -1679948.187
2047505.1 -1676110.68
2186917.593 -1518464.703
  Pen (1,2,0)
  Brush (0,1)
  Center 2140313.457 -1623802.989
```

- ❶ Start of the entry for the first territorial unit in our data section. This region definition will be associated to the identifier on the first entry of the MID file. The 2 parameter near Region shows that this region is made of two polygons (example, France may be considered as a region made of two polygons: metropol and Corse island).
- ❷ The first polygon of this region is set with seven points whose coordinates in X Y forms are given on following lines.

- ③ **Pen(a, b, c), Brush(a, b) and Center x y** specifications are optional and they will not be read by HyperAdmin.
- ④ The second polygon of this region is defined with five points whose coordinates are given on the five following lines.
- ⑤ Here is the start of a new **Region** definition. As the second entry of the data section, this region definition will be associated to the identifier on the second line of the MID file. **Region 1** indicates that this region is made of one polygon.
- ⑥ This line shows the number of points that compose the polygon: 11 points, whose coordinates are successively given on 11 following lines.

3.4. Layer of main cities

HyperAtlas can handle additional layers of information that can be displayed over the maps. Currently (May 2011), only a layer showing the main cities has been tested and can currently be supported.

The expected format for this "cities" layer incorporation into the dataset to be built is a **.csv** file. This file is only composed of three fields, these fields are separated by a comma character:

- the name of the city
- the X coordinate of this city, based on the MIF/MID projection and coordinates system
- the Y coordinate of this city, based on the MIF/MID projection and coordinates system

The following listing provides an example of the main cities layer definition csv file that has been used for European datasets (EPSG 3035):

```
Vilnius,5295673.924,3612560.328
Minsk,5460580.445,3560616.774
Dublin,3253284.971,3480193.09
Berlin,4547186.818,3272495.918
Amsterdam,3975886.565,3263689.867
Warszawa,5068508.328,3293815.926
London,3620060.313,3202333.12
Bruxelles/Brussel,3927032.583,3095975.903
Kyiv,5751996.553,3239855.146
Praha,4639737.703,3008973.669
Paris,3769691.587,2891825.057
Wien,4790135.661,2807741.98
Budapest,5003603.404,2753261.228
Bern,4128054.027,2651781.399
Beograd,5142183.84,2467117.484
Bucuresti,5593724.067,2506886.924
Sofiya,5408445.047,2274434.026
Tirana,5143864.946,2078891.927
Madrid,3164690.758,2032301.915
Ankara,6248076.399,2163898.451
Helsinki,5144699.201,4208069.911
Zagreb,4784474.809,2540154.601
Nicosia,6434072.209,1668719.112
Luxembourg,4054388.133,2965578.225
Bratislava,4859375.987,2822228.019
Tallinn,5154761.636,4105585.175
Sarajevo,4997878.051,2344715.534
Skopje,5274194.7,2172377.111
Athina,5518075.047,1777730.958
Kishinev,5733746.751,2835203.886
```


Geometry input

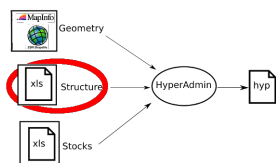
Copenhagen, 4481880.455, 3626362.309
Lisboa, 2671218.026, 1947183.08
Oslo, 4362362.69, 4091266.484
Reykjavik, 2843090.801, 4908517.82
Riga, 5170116.607, 3836021.74
Roma, 4531433.066, 2089563.772
Stockholm, 4781578.636, 4041161.089
Valletta, 4737055.11, 1442089.281
Ljubljana, 4670851.053, 2559186.916
El-Jazair, 3696198.974, 1536632.051
Tounis, 4344016.475, 1511814.733
Podgorica, 5085720.438, 2197200.507
Vaduz, 4287807.431, 2668956.206

Chapter 4. Structure/Contiguity input

The Section 4.1 presents the expectations of the HyperAdmin about the structure input file, e.g. the information about the territorial units hierarchy and their relationships.

The Section 4.2 presents the optional steps that consists in creating a distance-time contiguity matrix input data for custom neighbourhood definitions (example: distance time, 2 hours by car, etc.).

4.1. Structure input



In the the input `structure.xls` Excel file, ten sheets must mandatory be provided in a unique `.xls` file.

Optionnally, complex contiguities must be defined as a set of seven sheets in an unique other xls file. Thus the input data may be composed of:

- `some_structure.xls`: to describe the structure;
- `some_contiguity.xls`: to optionally describe the contiguities (see Section 4.2).

Table 4.1 provides the list of these sheets names and a short description for each of them, as they are expected in the `structure` input definition. The expected columns and an example for each of them is described below this table.



The names of sheets is case-unsensitive. For example, the mandatory `UnitArea.txt` file can be named `unitarea.txt` or `UNITAREA.txt`. Suffixes and prefixes around the file basename are also possible while choosing the plain text option: thus, `my_UnitArea.txt` will be considered as a `UnitArea.txt` file, e.g. the input file that contains the information about the relationships between areas and territorial units.

Table 4.1. Overview of expected sheets for data structure input

Excel Sheet Name	Description
unit	Identifiers for the set of territorial units. See Content of unit.
Area	Identifiers for the set of study areas. See Content of area.
Zoning	Identifiers for the set of meshes. See Content of zoning.
UnitSup	Hierarchy between units: an UTSup_ID parent unit owns at least one child UT_ID unit. See Content of unitsup.
UnitArea	An UT belongs to one or several study areas. See Content of unitarea.
UnitZoning	An UT belongs to one or several meshes. See Content of unitzoning.
language	Provides a human readable name for used languages codes. See Content of language.
UnitLanguage	Names of UT in different languages. A translation may be missing. See Content of unit-language.
AreaLanguage	Names of the study areas in different languages. A translation may be missing. See Content of arealanguage.
ZoningLanguage	Names of the meshes in different languages. A translation may be missing. See Content of zoninglanguage.

Expected content for each of these sheets is following:

Unit This file/sheet must contain one column whose header cell must be UT_ID.
Example:

Table 4.2. Sample input Unit sheet

UT_ID
AT11
AT12
AT13
AT21
etc...

Area This file/sheet must contain one column whose header cell must be Area_ID.
Example:

Table 4.3. Sample input Area sheet

Area_ID
UE15
UE25
PECO
Arc_Atlantique
Nouveaux_UE
UE27
UE29

Zoning

This file/sheet must contain one column whose header cell must be Zoning_ID. An additional column named Rank may order given zonings. Example:

Table 4.4. Sample input Zoning sheet

Zoning_ID	Rank
Nuts_0	1
Nuts_1	2
Nuts_2	3
Nuts_3	5
Nuts_2_3	4

UnitSup

This file/sheet must contain two columns whose header cells must be named UTSup_ID and UT_ID. Example:

Table 4.5. Sample input UnitSup sheet

UT_ID	UTSup_ID
AT1	AT
AT2	AT
AT3	AT
BE1	BE
BE2	BE

UnitArea

This file/sheet must contain two columns whose header cells must be named UT_ID and Area_ID. Example:

Table 4.6. Sample input UnitArea sheet

UT_ID	Area_ID
AT	UE15
BE	UE15
DE	UE15
DK	UE15
ES	UE15

UnitZoning

This file/sheet must contain two columns whose header cells must be **UT_ID** and **Zoning_ID**. Example:

Table 4.7. Sample input UnitZoning sheet

UT_ID	Zoning_ID
AT	Nuts_0
BE	Nuts_0
BG	Nuts_0
CH	Nuts_0

Language

This file/sheet must contain two columns whose header cell are **Language_ID** and **Language_NAME**. Example:

Table 4.8. Sample input Language sheet

Language_ID	Language_NAME
DE	allemand
CS	tchèque
DA	danois
ET	estonien
EN	anglais
ES	espagnol

UnitLanguage

This file/sheet must contain three columns whose header cells must be **UT_ID**, **UT_NAME** and **Language_ID**. Example:

Table 4.9. Sample input UnitLanguage sheet

UT_ID	Language_ID	UT_NAME
AT11	DE	BURGENLAND
AT34	DE	VORARLBERG
BE24	NL	VLAAMS BRABANT
BE25	NL	WEST-VLAANDEREN
BE31	FR	BRABANT WALLON
BE32	FR	HAUT

AreaLanguage

This file/sheet must contain three columns whose header cells must be **Area_ID**, **Language_ID** and **Area_NAME**. Example:

Table 4.10. Sample input AreaLanguage sheet

Area_ID	Language_ID	Area_NAME
UE15	FR	Union européenne des 15
UE25	FR	Union européenne des 25
PECO	FR	Pays d'Europe Centrale et Orientale

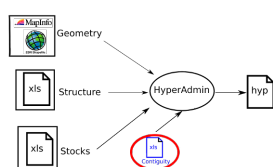
ZoningLanguage

This file/sheet must contain three columns whose header cells must be zoning_ID, Language_ID and zoning_NAME. Example:

Table 4.11. Sample input ZoningLanguage sheet

Zoning_ID	Language_ID	Zoning_NAME
Nuts_0	FR	Nomenclature des unités territoriales de niveau 0
Nuts_1	FR	Nomenclature des unités territoriales de niveau 1
Nuts_2	FR	Nomenclature des unités territoriales de niveau 2
Nuts_3	FR	Nomenclature des unités territoriales de niveau 3
Nuts_2_3	FR	Nomenclature des unités territoriales de niveau 2-3

4.2. Contiguity input (optional)



This section presents the optional contiguity definition input data file.

Table 4.12. Overview of expected sheets for contiguity input

Excel Sheet Name	Description
Contiguity	List of identifiers for contiguities.
ContiguityLanguage	Names of the contiguities in different languages. A translation may be missing. See Content of contiguitylanguage.
Neighbourhood	Unique code for a neighbourhood that is associated to a contiguity, a threshold and a comparator. The comparator shows if two UT are neighbours or not. See Content of neighbourhood.
Neighbourhood	Names of neighbourhoods for each language. A translation may be missing. See Content of neighbourhoodlanguage.
ContiguityZoning	A distance matrix is available for one or several meshes. See Content of ContiguityZoning.
ContiguityArea	A distance matrix is available for one or several study areas. See Content of ContiguityArea.
UnitContiguity _i	Each line provides the code of two UT, following columns show the distance for Contiguity_ID _i , the header cell of each contiguity column providing the identifier of this contiguity. See Content of UnitContiguity.

Following listing provides an example for each expected sheet describing a contiguity definition:

Contiguity.txt / Contiguity

This sheet must contain one column whose header cell is **ID**. Example: the following sample sets two possible computations for contiguity, distance-time matrixes will be available for a car and for a lorry.

Table 4.13. Sample input Contiguity sheet

ID
CAR
TRUCK

ContiguityLanguage.txt / ContiguityLanguage

This sheet must contain four columns whose header cells are **CONTIGUITY_ID**, **Language_ID** (see Note about expected languages identifiers), **Contiguity_NAME** (e.g. the name of this contiguity in this locale) and **Contiguity_DESC** (a description of this contiguity). Example:

Table 4.14. Sample input ContiguityLanguage sheet

Contiguity_ID	Language_ID	Contiguity_NAME	Contiguity_DESC
CAR	EN	car time	Time between units by car (in minutes)
TRUCK	EN	truck time	Time between units on a truck (minutes)
CAR	FR	temps voiture	Temps entre les unités en voiture (minutes)
TRUCK	FR	temps camion	Temps entre les unités en camion (minutes)

Neighbourhood.txt / Neighbourhood

This sheet must contain four columns whose header cells are **Neighbourhood_ID** (see Note about expected languages identifiers), **Contiguity_ID**, **Distance** and **Comparator**. Possible values for the **Comparator** cells are:

- <
- <=
- ==
- >=
- >

Example:

Table 4.15. Sample input Neighbourhood sheet

Neighbourhood_ID	Contiguity_ID	distance	comparator
CAR <= 360	CAR	360	<=
TRUCK <= 360	TRUCK	360	<=
CAR <= 540	CAR	540	<=
TRUCK <= 540	TRUCK	540	<=
CAR <= 180	CAR	180	<=
TRUCK <= 180	TRUCK	180	<=

**NeighbourhoodLanguage.txt /
NeighbourhoodLanguage**

This sheet must contain four columns whose header cells are Neighbourhood_ID, Language_ID (ISO-639 language in 2 digits, Neighbourhood_NAME (e.g the name of this neighbourhood) and Neighbourhood_DESC (e.g. a description of this neighbourhood). In the following example, neighbourhoods are translated in english (EN) and french (FR):

Table 4.16. Sample input
NeighbourhoodLanguage sheet

Neighbourhood	Language_ID	Neighbourhood	Neighbourhood_DESC
CAR <= 360	EN	6h car	Units at less than 6 hours by car
TRUCK <= 360	EN	6h truck	Units at less than 6 hours on a truck
CAR <= 540	EN	9h car	Units at less than 9 hours by car
TRUCK <= 540	EN	9h truck	Units at less than 9 hours on a truck
CAR <= 180	EN	3h car	Units at less than 3 hours by car
TRUCK <= 180	EN	3h truck	Units at less than 3 hours on a truck
CAR <= 360	FR	6h de voiture	Unités à moins de 6 heures en voiture
TRUCK <= 360	FR	6h de camion	Unités à moins de 6 heures en camion
CAR <= 540	FR	9h voiture	Unités à moins de 9 heures en voiture
TRUCK <= 540	FR	9h de camion	Unités à moins de 9 heures en camion
CAR <= 180	FR	3h de voiture	Unités à moins de 3 heures en voiture
TRUCK <= 180	FR	3h de camion	Unités à moins de 3 heures en voiture

ContiguityZoning.txt / ContiguityZoning

This sheet must contain two columns whose header cells are Contiguity_ID and Zoning_ID. In the following example, the distance-time by car and distance-time by lorry are available for NUTS_2 zoning only:

Table 4.17. Sample input ContiguityZoning sheet

Contiguity_ID	Zoning_ID
CAR	Nuts_2
TRUCK	Nuts_2

! The identifiers that are given in the Zoning_ID column must be coherent with the identifiers that have been given in the Zoning sheet of the structure input, see Content of zoning.

ContiguityArea.txt / ContiguityArea

This sheet must contain two columns whose header cells are Contiguity_ID and Area_ID. In the following example, distance-time matrixes by car are available for UE15 and UE25, distance-time matrixes by lorry are available for UE27 and UE29:

Table 4.18. Sample input ContiguityArea sheet

Contiguity_ID	Area_ID
CAR	UE15
CAR	UE25
TRUCK	UE27
TRUCK	UE29

! The identifiers that are given in the Area_ID column must be coherent with the identifiers that have been given in the Area sheet of the structure input, see Content of area.

UnitContiguity.txt / UnitContiguity_i

This sheet must contain at least three columns whose header cells are UT_ID1, UT_ID2, then the identifier of a contiguity.. In the following example, contiguities between units are performed for CAR contiguity and TRUCK contiguity:

Table 4.19. Sample input UnitContiguity sheet

UT_ID1	UT_ID2	CAR	TRUCK
ES12	UKC1	1265.11	1820.5
ES13	UKC1	1138.85	1649.5
ES21	UKC1	1058.92	1529.9

! In Excel mode, each UnitContiguity can only contain 2^{16} rows, e.g. 65536. Several sheets can be created to import more results: just name your sheets UnitContiguity₁, UnitContiguity₂, etc. Note that only 30 UnitContiguity_i sheets can be created. Nevertheless, on considering a symmetric relationship for a distance between two units (e.g. distance between UT1 and UT2 equals the distance between UT2 and UT1), the number of needed rows can quasi be reduced by half. Thus, the number S of needed sheets for n units can be found by executing the formula which is shown on Figure 4.1:

Figure 4.1. Number S of needed sheets for n units

$$S = \frac{1 + \frac{n \cdot (n-1)}{2}}{65536}$$

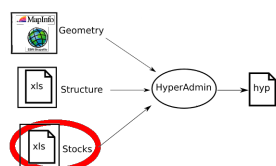
S is the number of needed sheets, n is the number of units. On the numerator, "1 +" stands for the header row that must be included on each sheet.

Note that **contiguities are not aggregable**: a distance matrix is set for a given level of mesh. For its upper level, the associated distance matrix must also be given. Table 4.20 provides an example of the number of needed sheets for different levels with several numbers of units.

Table 4.20. Example of needed sheets number

Mesh	Number of UT (n)	Number of sheets (S)
NUTS_0	29	1
NUTS_1	92	1
NUTS_2	280	2
NUTS_2_3	727	5
NUTS_3	1329	14
Total		23

Chapter 5. Stocks input



The **stock** file mainly aims at describing the statistics of the dataset.

5.1. HyperAdmin input data file format

This section describes the stocks (statistical data) file that HyperAdmin expects as input.



Please note the following requirements for the input data file:

- the input data file must be a spreadsheet xls file (editable by Microsoft Excel and Open Office) named "`*data*.xls`": the filename must include the "data" characters sequence and the .xls extension is required.
- the values of stocks must be provided for the lowest level of territorial units. This list is available in the example data template that depends on the selected structure/geometry model at previous step.
- all values for all units must be filled;

Following sections describe the expected format (sheets, columns and possible values) for the version 2 of this `data.xls` "stocks" file.

5.1.1. About

Table 5.1 provides an example for this mandatory sheet in the data v2 input xls file.

Table 5.1. V2 sample About sheet

VERSION	TIME_ENABLED
2	TRUE

This sheet aims at identifying the version of the format of this data file. Currently (2010-2011), only the value **2** is possible for the VERSION column.

The expected value for the TIME_ENABLED column is a boolean: only **TRUE** or **FALSE** values are possible:

- The **TRUE** value shows that values are available for the same labels of indicators at several dates: for example, the population in 2000, the population in 2002.
- The **FALSE** value shows that each indicator is given for a single date.

5.1.2. Data

Table 5.2 provides an example for this mandatory sheet in the data v2 input xls file.

Table 5.2. V2 sample Data sheet

UT_ID	pop2000	pop2002	area2000	gdp2000	gdp2002
AT111	1	15	2	7	10
AT112	3	16	4	8	11
AT113	5	17	6	9	12

This sheet must provide at least three columns: UT_ID then at least two indicators identifiers (in HyperAtlas, there must be at least one numerator stock and one denominator stock). The Table 5.2 shows five indicators identifiers: pop2000, pop2002, area2000, gdp2000 and gdp2002. These identifiers must be described in the **StockInfo** sheet (see Section 5.1.8).

The UT_ID column must provide the list of territorial units at the lowest rank (example, at NUTS 3 level) of the dataset. The units are referenced by their identifiers that must match the given values in the associated `structure.xls` input file.

Then, each other cell provides a value for the given indicator column at the given unit row. For example in Table 5.2, 17 is the value for pop2002 indicator in AT113 territorial unit.

 Each cell must be valuated. Missing values are not accepted here.

5.1.3. Default

Table 5.3 provides an example for this optional sheet in the data v2 input xls file.

Table 5.3. V2 sample Default sheet

DEFAULT_NUM	DEFAULT_DEN
pop	area

This sheet aims at providing a default indicator to be selected in HyperAtlas at startup for the denominator and for the numerator combo boxes. Expected values for both columns are valid indicators identifiers that must match two of those defined in the **StockInfo** sheet (see Section 5.1.8).

5.1.4. Label

Table 5.4 provides an example for this mandatory sheet in the data v2 input xls file.

Table 5.4. V2 sample Label sheet

LABEL_ID	LANG_CODE	NAME	DESC
1	EN	Total population	Total population in thousands
1	FR	Population totale	Population totale en milliers
2	EN	Area	Total area
2	FR	Superficie	Superficie totale
3	EN	GDP	Gross domestic product
3	FR	PIB	Produit intérieur brut
4	EN	GDP/Inhabitant	Gross domestic product per inhabitant
4	FR	PIB/Hab	PIB par habitant
5	EN	Density	Density of population
5	FR	Densité	Densité de population

This sheet aims at providing the internationalized names and descriptions for the indicators and pre-defined ratios. The `LABEL_ID` and `LANG_CODE` provides indexes for this table: for a given label identifier there may be several available translations. Thus, the `LABEL_ID = 1` is available in english (`LANG_CODE = EN`) and french (`LANG_CODE = FR`) languages. In the **StockInfo** sheet, each indicator reference a label identifier. As several indicators may be similarly named and described (when an indicator is valuated for several dates), these labels have been exported here.



The language identifier code must be a valid ISO Language Code. These codes are the lower-case, two-letter codes as defined by ISO-639. Nevertheless, the parser supports upper-cases. You can find a full list of these codes at a number of sites, such as: <http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt> (2011-03-16).

Note that values in the `LABEL_ID` column may be referenced from the **StockInfo** sheet (see Section 5.1.8) and from the **RatioStock** sheet (see Section 5.1.7).

5.1.5. Metadata

Table 5.5 provides an example for this optional sheet in the data v2 input xls file.

Table 5.5. V2 sample Metadata sheet

UT_ID	STOCK_ID	PROVIDER_ID
AT111	pop2000	1
AT112	pop2000	2
	area	2
	pop2002	1

This draft sheet aims at providing some basic metadata information for an indicator relatively or not to a territorial unit. Currently, only the source of data may be given as metadata.

For example in Table 5.5, the values of the `pop2000` indicator identifier were retrieved from different sources for regions `AT111` and `AT112`. On the contrary, all values for the `area` indicator, whatever the unit is, were provided by the same source. Idem for the `pop2002` indicator.

The values in the `PROVIDER_ID` column must match the identifiers that are given in the **Provider** sheet (see Table 5.6). Likewise, the values in the `STOCK_ID` column must match the identifiers that are defined in the **StockInfo** sheet (see Table 5.8).

5.1.6. Provider

Table 5.6 provides an example for this optional sheet in the data v2 input xls file.

Table 5.6. V2 sample Provider sheet

PROVIDER_ID	NAME	CONTACT	URL
1	Eurostat	toto@eurostat.eu	http://www.eurostat.eu
2	INSEE	tata@insee.fr	http://www.insee.fr

This sheet aims at providing the list of data providers. Their different ids are referenced from the **Metadata** sheet.

5.1.7. RatioStock

Table 5.7 provides an example for this optional sheet in the data v2 input xls file.

Table 5.7. V2 sample RatioStock sheet

RATIO_ID	LABEL_ID	NUM_ID	DEN_ID	VALIDITY_START	VALIDITY_END
1	4	gdp2000	pop2000	2000	2000
2	4	gdp2002	pop2002	2002	2002
3	5	pop2000	area2000	2000	2000
4	5	pop2002	area2000	2002	2002

This sheet aims at defining relevant ratios for the HyperAtlas "ratio" combo box parameter. Table 5.7 shows the example of two such predefined ratios, each of them for two different dates:

- the GDP/Inhabitant:
 - in 2000 (second line)
 - in 2002 (third line)
- The density of population:
 - in 2000 (fourth line)
 - in 2002 (fifth line)

Each value in the RATIO_ID column must be unique. Doublons will overwrite the previous found value.

Note that the LABEL_ID references the same labels for the given pairs of numerator/denominator at different dates (4 for lines 2 and 3, 5 for lines 4 and 5). These labels identifiers must be set in the **Label** sheet (see Section 5.1.4).

The values in the NUM_ID column and the values in the DEN_ID column must match the identifiers of indicators that are defined in the **StockInfo** sheet (see Section 5.1.8).

The values in the VALIDITY_START column will only be considered if the value of the TIME_ENABLED column in the **About** sheet is TRUE (see Section 5.1.1). Then, one relevant ratio can be chosen in HyperAtlas for different dates. Identically for the values in the VALIDITY_END column. Though VALIDITY_START and VALIDITY_END columns are designed to handle time intervals, setting the same value in both columns makes the ratio associated to a timestamp.



The expected format for both validity start/end date fields is currently a year in the yyyy pattern. An input like 2010-12-31 is possible but this version of the application will only take into account the year, that is to say 2010 for this example.

5.1.8. StockInfo

Table 5.8 provides an example for this mandatory sheet in the data v2 input xls file.

Table 5.8. V2 sample StockInfo sheet

STOCK_ID	LABEL_ID	MEASURE_UNIT	VALIDITY_START	VALIDITY_END	VISIBLE_FLAG
pop2000	1	*1000	2000	2000	TRUE
pop2002	1	*1000	2002	2002	TRUE
area2000	2	km2	2000	2000	TRUE
gdp2000	3	euros	2000	2000	TRUE
gdp2002	3	euros	2002	2002	TRUE

Stocks input

This sheet mainly aims at providing the identifiers of the indicators of the dataset. Here are a short description for each column of this sheet:

- **STOCK_ID**: each value in this column must be unique. Any doublon will overwrite the previous found identical value. This column lists the identifiers of the indicators that are referenced in the other sheets. Note that several indicators may be associated to the same label (lines 2 and 3 for example), though they exist to distinguish the values of the population in 2000 and 2002.
- **LABEL_ID**: each value in this column must reference an identifier defined in the **Label** sheet (see Section 5.1.4).
- **MEASURE_UNIT**: simply provides the unit of measure for this indicator.
- **VALIDITY_START**: shows the start date of validity for this indicator. This field will only be considered if the value of the **TIME_ENABLED** column in the **About** sheet is **TRUE** (see Section 5.1.1 and Important note about expected date format).
- **VALIDITY_END**: shows the end date of validity for this indicator. **VALIDITY_START** and **VALIDITY_END** fields are able to manage time intervals, but they can be used to associate a time-stamp to the current stock: just write the same value in both cells (please see Important note about expected date format).
- **VISIBLE**: this field acts like a flag, a boolean is expected for the values of this column. A **TRUE** value shows that this indicator will be available in the numerator and in the denominator combo boxes of HyperAtlas parameters panel. A **FALSE** value may be usefull to define relevant ratios whose indicators have no reason to be available in the numerator and denominator combo boxes. For example, the life expectancy pre-defined ratio considers indicators that have no sense out of this compute.

Appendix A. References

Links

- [1] ESPON Coordination Unit. *ESPON*. [on line]. <http://www.espon.eu/> (last visit: 1.st december 2009).
- [2] ESRI. *ESRI Shapefile Technical Description - An ESRI White Paper - July 1998*. [on line]. <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> (last visit: 28th of July 2015).
- [3] HyperCarte Research Group. *Site Web du Groupe de Recherche HyperCarte - Accueil*. [on line]. <http://hypercarte.imag.fr> (last visit: 28th of July 2015).
- [4] Pitney Bowes MapInfo. *Appendix J: MapInfo Data Interchange Format*. [on line]. http://resource.mapinfo.com/static/files/document/1074660800077/interchange_file.pdf (last visit: 28th of July 2015).
- [5] Politiques de la Terre. *Présentation*. [on line]. http://politiquesdelaterre.fr/?page_id=2 (last visit: 28th of July 2015).
- [6] QGIS. *QGIS - A Free and Open Source Geographic Information System*. [on line]. <http://www.qgis.org/en/site/> (last visit: 28th of July 2015).
- [7] Wikipedia. *ESRI Shapefile*. [on line]. <https://en.wikipedia.org/wiki/Shapefile> (last visit: 28th of July 2015).

About

This document has been generated on the 2015-08-03 11:35:36, from sources of the forge imag hypercarte [<https://forge.imag.fr/projects/hypercarte/>] project at the svn rev 828.

For any comment question or suggestion, please visit <http://hypercarte.imag.fr> or send an email to [<hypercarte@imag.fr>](mailto:hypercarte@imag.fr).

Other contact: Jérôme Gensel ([<Jerome.Gensel@imag.fr>](mailto:Jerome.Gensel@imag.fr)) available at the following address:

Laboratoire LIG UMR 5217, Equipe STeamer
681 rue de la Passerelle, Domaine Universitaire
BP 72
38402 Saint Martin d'Hères Cedex
FRANCE
Tel: (+33) (0)4 76 82 72 80
Fax: (+33) (0)4 76 82 72 87



Colophon

Based on DocBook technology ¹, this document is written in XML format, sources are validated with DocBook DTD 4.5CR3, then sources are transformed to HTML and PDF formats by using DocBook xslt 1.73.2 stylesheets. The generation of the documents is automatized thanks to the docbench tool that is based on Ant ², java ³, processors Xalan⁴ and FOP ⁵. Note that Xslt standard stylesheets are customized in order to get a better image resolution in PDF generated output for admonitions icons: the generated sizes of these icons were turned from 30 to 12 pt.

¹[on line] *DocBook.org* [<http://www.docbook.org>] (last visit: July 2011)

²[on line] *Apache Ant - Welcome*. Version 1.7.1 [<http://ant.apache.org>] (last visit: July 2011)

³[on line] *Developer Resources For Java Technology* [<http://java.sun.com>] (last visit: July 2011). Version 1.6.0_03-b05.

⁴[on line] *Xalan-Java Version 2.7.1* [<http://xml.apache.org/xalan-j/>] (last visit: 18 november 2009). Version 2.7.1.

⁵[on line] *Apache FOP* [<http://xmlgraphics.apache.org/fop/download.html>] (last visit: July 2011). Version 0.94.